2013-09

Unsupervised topic discovery by anomaly detection

Cheng, Leon

Monterey, California: Naval Postgraduate School

NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

UNSUPERVISED TOPIC DISCOVERY BY ANOMALY DETECTION

by

Leon Cheng

September 2013

Thesis Co-Advisors: Craig Martell
Pranav Anand

Approved for public release; distribution is unlimited

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE September 2013	3. REPORT TYPE AND DATES COVERED Master's Thesis

4. TITLE AND SUBTITLE UNSUPERVISED TOPIC DISCOVERY BY ANOMALY DETECTION	5. FUNDING NUMBERS
6. AUTHOR(S) Leon Cheng	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000	**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A	**10. SPONSORING/MONITORING AGENCY REPORT NUMBER**

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government. IRB Protocol number ____N/A____.

12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited	12b. DISTRIBUTION CODE A

13. ABSTRACT (maximum 200 words)

With the vast amount of information and public comment available online, it is of increasing interest to understand what is being said and what topics are trending online. Government agencies, for example, want to know what policies concern the public without having to look through thousands of comments manually. Topic detection provides automatic identification of topics in documents based on the information content and enhances many natural language processing tasks, including text summarization and information retrieval. Unsupervised topic detection, however, has always been a difficult task. Methods such as Latent Dirichlet Allocation (LDA) convert documents from word space into document space (weighted sums over topic space), but do not perform any form of classification, nor do they address the relation of generated topics with actual human level topics. In this thesis we attempt a novel way of unsupervised topic detection and classification by performing LDA and then clustering. We propose variations to the popular K-Mean Clustering algorithm to optimize the choice of centroids, and we perform experiments using Facebook data and the New York Times (NYT) corpus. Although the results were poor for the Facebook data, our method performed acceptably with the NYT data. The new clustering algorithms also performed slightly and consistently better than the normal K-Means algorithm.

14. SUBJECT TERMS Unsupervised topic detection, Anomaly detection, K-means clustering, Latent Dirichlet Allocation	15. NUMBER OF PAGES 67
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UU

NSN 7540-01-280-5500 Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release; distribution is unlimited

UNSUPERVISED TOPIC DISCOVERY BY ANOMALY DETECTION

Leon Cheng
Civilian, Defence Technology and Agency, Singapore
B.Eng., National University of Singapore, 2003

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

from the

NAVAL POSTGRADUATE SCHOOL
September 2013

Author: Leon Cheng

Approved by: Craig Martell
Thesis Co-Advisor

Pranav Anand
Thesis Co-Advisor

Peter J. Denning
Chair, Department of Computer Science

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

With the vast amount of information and public comment available online, it is of increasing interest to understand what is being said and what topics are trending online. Government agencies, for example, want to know what policies concern the public without having to look through thousands of comments manually. Topic detection provides automatic identification of topics in documents based on the information content and enhances many natural language processing tasks, including text summarization and information retrieval. Unsupervised topic detection, however, has always been a difficult task. Methods such as Latent Dirichlet Allocation (LDA) convert documents from word space into document space (weighted sums over topic space), but do not perform any form of classification, nor do they address the relation of generated topics with actual human level topics. In this thesis we attempt a novel way of unsupervised topic detection and classification by performing LDA and then clustering. We propose variations to the popular K-Mean Clustering algorithm to optimize the choice of centroids, and we perform experiments using Facebook data and the New York Times (NYT) corpus. Although the results were poor for the Facebook data, our method performed acceptably with the NYT data. The new clustering algorithms also performed slightly and consistently better than the normal K-Means algorithm.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

LIST OF FIGURES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

I want to acknowledge my wife, Yvonne, for the support she gives me by taking care of our household and giving me the peace of mind to pursue my interest in researching this thesis. I also want to thank Professor Martell and Professor Anand for their invaluable time, patience and advice in guiding me throughout the course.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

A. BACKGROUND

In this digital age, there is a large supply of information that is easily and conveniently reachable by everyone. Coupled with the explosion of social media usage, people from all walks of life are making their opinions and sentiments regarding information they read known publicly. There is a strong interest in the analysis of these opinions and comments as they provide useful information about the sentiments and the concerns about a particular issue or product.

Research in topic detection started in 1998 as part of Topic Detection and Tracking (TDT) [1] under the DARPA Translingual Information Detection, Extraction, and Summarization (TIDES) program. Topic detection refers to the ability to automatically identify topics in documents based on the information content. Topic detection benefits many natural language processing tasks, including text summarization and information retrieval. This is an essential step towards the building of a system that provides an efficient way for analysts to seek required information from a vast supply of unlabeled data.

1. Supervised, Unsupervised and Semi-Supervised Learning Methods

Supervised learning refers to a machine learning task where the machine learns from a set of labeled training data in order to label new data. Training data refers to a set of data that consists of input vectors and their corresponding labels. In the field of topic detection, a training document would consist of a text document and the correct topic it belongs to. Unsupervised learning refers to a machine learning task where the machine does not have a set of labeled training data to learn from, and has to label new data by finding hidden patterns or structure within the new data itself.

In many data mining and machine learning tasks, there is a large supply of unlabeled data and only a small amount of labeled data due to the high cost of generating labeled data. This makes supervised learning a potentially expensive task. This also

serves as an impetus to explore unsupervised or semi-supervised means of topic detection.

Unsupervised topic detection has always been a difficult task. One of the key tasks of unsupervised topic detection is to derive an algorithm to find a set of keywords that describes a document. By calculating the probability of the co-occurrence of certain word groups, we can identify possible topics. Another method to derive an algorithm could be by looking at a document and trying to determine its keywords by analyzing the structure of the document. These are some of the challenges of unsupervised topic detection.

Semi-supervised learning methods require a small amount of labeled data to aid the unsupervised learning. To complement semi-supervised methods, active learning methods enable the user to feed information back into the system as it is in the process of labeling to aid in its accuracy. In the field of topic detection, an active learning method could be to ask the user for the topic a selected document belongs to. The machine then uses that information to help find the hidden structure in the document.

2. Social Media Content

For part of the experiments, we are using data obtained from a Facebook page concerned with a whitepaper on population [2] released by the Singapore government. The data is extracted from the comments about the whitepaper, and it is written primarily in Singlish. Singlish has its roots in the English language, but with an additional vocabulary of words from other languages such as Malay, Tamil, Mandarin and other Mandarin dialects such as Hokkien and Teow Chew.

Unlike formal documents social media content tends to be shorter and less coherent topically. Very often, authors of social media content just wish to express their sentiments towards a topic, and they do not need their readers to understand the full context of why they said it. The comments are often emotional rather than reasoned and supported by research. As a result, social media content may consist of comments or posts that have nothing to do with the topic that inspired it. For example, in our

experiments, we found many comments are pure insults and other comments, while not insulting, have no relevance to the content of the whitepaper.

B. RESEARCH APPLICATION

With the vast amount of information available online, it is of increasing interest to understand what is being said, what the areas of concern are and what topics are trending online. For example, it would be useful for Government agencies to find out what areas of concerns the public may have on certain issues or the popularity of certain policies without having to look through thousands of comments manually.

C. RESEARCH QUESTION AND PROPOSED SOLUTION

Our research question was, "Given data from a political forum page or any other social media page, can unsupervised topic detection techniques accurately detect topics that are of concern in the forum?"

We attempt to answer this question by applying known methods to convert the data set from unigram word space to that of topic space. In other words, instead of describing a document by the probability of occurrence of its words, we will describe a document as a probability of possible topics. This effectively reduces the dimensionality of the data set. It also allows us to deal with topic space in which we are interested. We then use clustering techniques to cluster similar data points together. We then look for the data outliers and create new clusters based on these outliers. We propose new variations of the K-means clustering algorithm, which allows the number of clusters to grow from one to K and then beyond K. The clustering will stop when either a specified maximum number of clusters has been reached or if there are no more outliers.

D. ORGANIZATION OF THESIS

In order to investigate the research question, this thesis is organized as follows:

- Chapter I discusses the background and motivation of this thesis. It also introduces the methods used in the research.
- Chapter II discusses prior and related work in the fields of topic detection.
- Chapter III contains a description of the methods used to prepare the data and to conduct the experiment.
- Chapter IV contains the results and analysis of the experiment.
- Chapter V contains the conclusion and recommendations for possible future work.

II. PRIOR AND RELATED WORK

A. PRIOR WORK

Detecting topics in a large collection of unstructured text without any prior knowledge or understanding of the potential underlying topics remains a difficult problem to solve. Latent Dirichlet Allocation (LDA) [3] is a generative model that discovers words that often appear together in different documents and groups them together as a topic. The problem with using generic LDA to discover topics is that it often generates bad topics that do not make any sense to the user. Despite its shortcomings, LDA's strength is in the fact that it is able to detect topics in an unsupervised manner. Hence, the experiments discussed in this thesis will use LDA as a starting point for converting the documents from word space into topic space.

Other work done in this field has involved detecting topics by clustering keywords that are extracted from the documents. Wartena *et al.* [4] discussed a method that involves extracting informative keywords which best describe the documents and then clusters the documents based on various similarity measures and represents them as topics. The difficulty in this approach is finding a good set of keywords that accurately represents the documents. The method used to discover keywords often includes selecting the most frequent terms in the documents and filtering out stop words or overly general terms. Conceptually, this method of detecting topics by clustering keywords is related to probabilistic latent semantic analysis (PLSA) [5], which models the co-occurrence of words and documents as a mixture of conditionally independent multinomial distributions.

1. Anomaly Detection

Anomaly detection is a common but important problem that has been researched extensively in various domains. In the domain of topic detection in text data, finding anomalous patterns in data can be interpreted as detecting new topics. Various techniques have been used to detect anomalous topics in text data such as mixture of models [5], statistical profiling with histograms [6, 7, 8], support vector machines (SVM)

[9], neural networks [10] and clustering [11]. These methods are described in more detail in the following subsections.

a. Mixture of Models

Baker *et al.* [5] described a novel way to detect and track topics based on probabilistic, generative models. In general, this technique uses a mixture of parametric statistical distributions to model the data. Assumptions are made regarding the distribution from which the data is generated. For example, we can assume normal data is generated from a Gaussian distribution, and anomalous data is generated from another Gaussian distribution but with different parameters (mean and variance). Grubb's test is then applied on a test instance and subsequently labeled accordingly. Another method related to this technique is to make assumptions about the normal data using different distribution parameters. Anomalous data is detected when a test instance does not fall into the assumed distributions. The difficulty in such a technique is finding the best fit distribution and parameters for the data set. There needs to be some sort of prior knowledge of the data set in order to assume the best distribution and parameters from which to generate the data.

b. Statistical Profiling with Histograms

The use of histograms to detect anomalies is very popular in the fields of intrusion detection [6, 7] and fraud detection [8] because the data is usually governed by a certain software or system profile. This is a nonparametric technique in that no assumptions are made a priori about the given data set. The model structure is instead determined from the given data set. Using histograms is an example of a simple method to profile the given data. Detecting anomalies using histograms first involves building a histogram based on the training data set. The technique then checks if a particular test instance falls into any one of the bins of the histogram. The test instance is then labeled as anomalous if it fails to fall into any one of the bins of the histogram. A critical aspect of this technique is determining the optimal size of the histogram bins. A large bin will result in many anomalous data instances being labeled as normal and a small bin may result in many normal data instances being labeled as anomalous.

c. Support Vector Machines

SVM was first introduced by Cortes and Vapnik [9], and it was used to detect anomalies in a single class setting. The basic idea of SVM is to construct an optimal hyper plane for linearly separable data patterns. SVM constructs a hyper plane that separates the normal from the anomalous data given a training data set. The hyper plane is chosen by maximizing the distance from the decision vectors (vectors closest to the decision boundary). SVM can also be extended to non-linearly separable data patterns by the usage of a kernel function. A kernel function basically transforms the original data into a new space which can be separated by a hyper plane. The SVM will then detect anomalous data by determining to which region a test data instance belongs. Hilt and Merat [10] used non-parametric SVM clustering to perform classification in an unsupervised setting and achieved improved performance over other parametric methods. That was done by accessing the SVM confidence parameters and switching the labels of the lowest confidence data points before performing SVM again. Repetition of this process improved the accuracy of classification.

d. Clustering Based

Clustering is a very popular technique and has many applications in a wide range of fields [11]. Clustering is generally an unsupervised technique which groups similar data instances into clusters. Since it tries to group similar data instances together, it would also have the inherent ability to sift out data points that are ‘not similar’ or anomalous to a cluster. Clustering based anomaly detection techniques can be grouped into three different categories relying on three different assumptions about what it means to be an outlier.

(1) Anomalous data do not belong to any clusters. There are clustering algorithms that do not require a data point to belong to any cluster. Such algorithms will sift out data points that do not belong to any cluster and label them as anomalies.

(2) Normal data points lie closest to their cluster centroid whereas anomalous data points lie furthest away. The anomaly detection technique uses

a distance measure from the closest centroid as the data instances' anomaly score. This technique fails if the anomalous data points form a cluster by themselves.

(3) Normal data points belong to large and dense clusters. Anomalous data points belong to small or sparse clusters. Techniques relying on this assumption declare data points belonging to clusters whose size or density falls below a pre-determined threshold as anomalous.

In our experiments, we will be using a clustering based approach due to its ability to operate unsupervised. We will be relying on the second method to locate outliers in a cluster. Additionally, we make some changes to the algorithm to allow us to detect anomalous data points even if they form a cluster by themselves.

2. K-Means Clustering

The K-means clustering algorithm [12] groups the data set into K disjoint clusters. It does this by assigning K cluster centroids and assigning data points to the cluster that is associated with its nearest centroid. The centroid is usually calculated as the arithmetic mean of the points in its cluster. Once all the points are assigned to a cluster, we have the initial clusters. At this point, the centroids will need to be recalculated and all the data points reassigned according to the new centroids calculated. The process of centroid recalculation and data point reassignment is repeated until the centroid does not change or the data points do not get reassigned anymore. At this point, the clustering is completed. In summary, the K-means clustering algorithm is composed of the following steps:

Step 1. Randomly select K data points as cluster centroids.

Step 2. Assign each data point to a cluster with the nearest centroid.

Step 3. After all data points are assigned, recalculate the cluster centroids.

Step 4. Repeat step 2 and 3 until the centroids no longer move or the data points are no longer reassigned.

a. Distance Measurement

There are several ways to compute distances between data points and the centroid. These can also be considered as similarity measures. The most common method of calculating distance is using the Euclidean distance. The Euclidean distance is the 'ordinary' distance between the two data points, A and B, and is given in by Equation 2.1.

$$\text{EuclideanD(X,Y)} = \sqrt{\sum_{i=1}^{n} (X_i - Y_i)^2} \tag{2.1}$$

Another way to measure the similarity between data points A and B is by measuring the cosine similarity. Cosine similarity between A and B is given in Equation 2.2.

$$\text{CosineSimilarity(X,Y)} = \frac{\sum_{i=1}^{n} X_i \times Y_i}{\sqrt{\sum_{i=1}^{n} (X_i)^2} \times \sqrt{\sum_{i=1}^{n} (\text{Y}_i)^2}} \tag{2.2}$$

We can convert this cosine similarity into a distance measure by subtracting it from one as shown in Equation 2.3.

$$\text{CosineDistance(X,Y)} = 1 - \text{CosineSimilarity(X,Y)} \tag{2.3}$$

The Kullback-Leibler (KL) divergence [13] measures the difference between two different probability distributions. This measure of distance can be used if the data points reflect probability distributions. The KL divergence of B from A is given in Equation 2.4.

$$D_{KL}(\text{P} \,||\, \text{Q}) = \sum_{i=1}^{n} \ln\left(\frac{P(i)}{Q(i)}\right) P(i) \tag{2.4}$$

In our experiments, we used KL divergence as a distance measure because the data points are tuples of probabilities; each document is in effect a probability distribution of topics (as a result of using LDA to reduce each document to a weighted sum of topics).

3. K-means Clustering with Constraints

Basu *et al.* [14] presented a way to implement K-means clustering with constraints. The fundamental idea was to build pairwise *must-link* and *cannot-link* constraints between points in the data set. An associated cost for violating each constraint could also be added. By specifying all the data points that must be linked, the initial clusters can be discovered by taking the transitive closure of the links. For example, if data points A and B must be connected and data points B and C must be connected, by transitivity, data points A and C should be connected as well. A, B and C would belong to the same neighborhood.

Since the generic K-means clustering algorithm cannot handle these pairwise constraints explicitly, the objective function has to be changed to take these constraints into consideration. The goal of clustering should now be minimizing the sum of the total distance between the points and their cluster centroids and the cost of violating the pairwise constraints. This means that when deciding which cluster to assign a data point, we do not just choose the cluster whose centroid is the nearest to the data point. We also have to take into consideration the constraints that are placed on that data point and the cost of violating the constraint. The objective function has to decrease with each cluster assignment until convergence.

Having the ability to apply constraints to the clustering algorithm allows active learning to be incorporated into the clustering algorithm to improve the clustering accuracy. Our method of constrained clustering is similar to the method proposed by Basu *et al.* except that we do not have any active learning in our experiments; we constrained clusters such that they do not consist of outliers, which we hypothesize as belonging to another cluster. Ensuring outliers do not exist in a cluster is similar to the idea of placing a *cannot-link* constraint between the outlier and the data points of the cluster.

In our experiments, we will propose two algorithms that enhance K-means clustering by using outlier constraints.

4. Detecting Cluster Outliers

Kriegel et al. [15] described a basic model to detect outliers by specifying a radius ε and a percentage π. A data point p is considered as an outlier if less than π percent of all other points have a distance to p greater than ε. A set of outliers can be described by Equation 2.5.

$$OutlierSet(\varepsilon, \pi) = \left\{ p \mid \frac{|dist(p,q) > \varepsilon|}{|\text{datapoints}|} < \pi \right\} \tag{2.5}$$

In our experiments, we will also use a distance based model to detect outliers. We define data points that are more than a specified number, α of standard deviations away from the mean as outliers. We have the ability to adjust α to obtain the best results.

5. Reducing Dimensionality

Kriegel *et al.* [16] suggested that clustering high-dimensional data presents some difficulty. It is difficult to visualize a high dimensional data set. There is an exponential growth of possible values with every dimension, and it is impractical to completely enumerate all possibilities. Another problem is that as dimensionality grows, the distance between data points converges, and hence, distance becomes increasingly imprecise. Very often, with high dimensional data, many of the attributes or features may be correlated or not relevant or meaningful for clustering.

A common method to reduce the dimensionality of a dataset introduced by Hotelling is known as Principal Component Analysis (PCA) [17]. PCA is a linear algebraic function that uses an orthogonal transformation whose axes are oriented in the direction of maximum variance of the data. The variance is maximum along the first axis of the new basis, while the second axis will maximize variance subject to the first axis orthogonally, and so forth. Dimensions can be reduced by rejecting the coordinates that correspond to the direction of minimum variance. The problem with PCA is that it is very sensitive to the scaling of the data. If the difference between two dimensions is huge, PCA would not be very useful. Usually, normalizing the data before applying PCA

would help. Ding *et al.* [18] described using PCA to discover the principal components that are used as the features in K-means clustering.

Another method to help reduce the dimensionality of a dataset is LDA. LDA converts the data set from its original feature space to that of a topic space whose dimensionality can be specified. We will be using LDA in our experiments to reduce the data dimensionality because we want to detect topics.

6. Latent Dirichlet Allocation

LDA is an unsupervised, generative probabilistic model of a corpus. Each of the documents in the corpus comprises random mixtures of latent topics, and each topic is described as a distribution of words. The LDA model discovers the latent topics by observing the words in the document. Once the generative procedure is established, we can define its joint distribution and use statistical inference to calculate the probability distribution over the latent topics, conditioned on the observed words. Documents are generated in the following steps:

1. Choose a K-dimensional topic weight vector θ_m from the distribution $P(\theta \mid \alpha)$ = Dirichlet(α)

2. For each word in the document:
   i. Choose a topic $z_n \in \{1 \ldots K\}$ from the multinomial distribution $P(z_n=k \mid \theta_m) = \theta_m^k$.
   ii. Given z_n, choose a word w_n from the probability distribution $P(w_n=i \mid z_n=j, \beta) = \beta_{ij}$.

The generative process above defines a joint distribution for each document. Given α and β, the joint distribution of a topic mixture θ, a set of N topics z, and a set of N words w is described in Equation 2.6.

$$P(\theta, z, w \mid \alpha, \beta) = p(\theta \mid \alpha) \prod_{n=1}^{N} P(z_n \mid \theta) P(w_n \mid Z_n, \beta) \tag{2.6}$$

It can also be diagrammatically represented as shown in Figure 1.

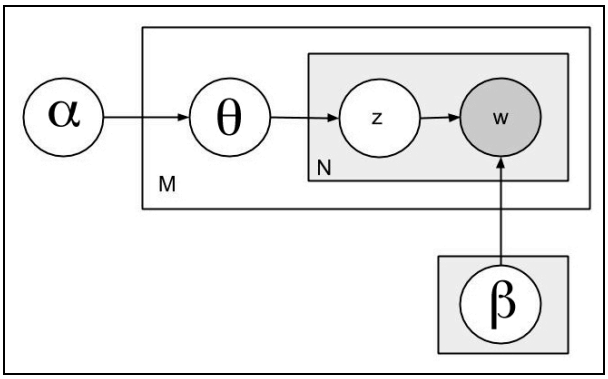

Figure 1. LDA

With LDA, we are able to find θ, the topic distribution for every document in the corpus. We will be using this probability distribution as input for our clustering algorithm.

B. CONCLUSION

For my experiment, we have decided to use LDA to reduce the dimensionality of the data because LDA changes the data set from word space into topic space, which we are interested in. We will then perform clustering on the documents using the generic K-means clustering algorithm and variations of the K-means clustering algorithm. Clustering was chosen as the method to classify the documents because of its ability to execute efficiently and without supervision.

THIS PAGE INTENTIONALLY LEFT BLANK

III. EXPERIMENT SETUP

A. SOURCE OF DATA

Two different sources of data were used for the experiments. One of the sources was from the popular *New York Times* (NYT) corpus [23] and the other source was from the comments left on a Facebook page that was addressing a particular political issue in Singapore. While the former was a well-recognized annotated data source, the latter was manually annotated by us.

The NYT annotated corpus consists of 1.8 million articles spanning 1987 to 2007. The articles are represented as XML files with the main article contained within the body tag of the XML file. The category to which the article belongs is contained in the meta tag named *Online_Sections*.

The data from Facebook is extracted from the comments on the Facebook page which addressed a population study whitepaper released by the Singapore government. We wanted to find out the areas which concerned the commenters most. Unlike the NYT corpus, the data here was mostly written in Singlish, and each comment is not as long as an article in the NYT. The documents that we were trying to classify here are typical of documents found in social media. These documents are usually short and use informal language.

The data sources were specifically chosen as we wanted to find out if the algorithms used will successfully detect topics unsupervised with these very distinct data sets.

B. DATA SELECTION AND PREPARATION

1. *New York Times* Corpus

The 1.8 million articles in the NYT corpus are categorized into 27 different categories. These categories are listed in Table 1.

Arts	Automobile	Books
Business	Corrections	Dining and Wine
Education	Front Page	Health
Home and Garden	Magazine	Movies
New York and Region	Obituaries	Opinion
Paid Death Notices	Real Estate	Science
Sports	Style	Technology
Theatre	Travel	U.S.
Washington	Week in Review	World

Table 1. 27 categories of NYT articles.

After reviewing the categories, we decided to omit the categories that could include other categories. For example, the category *Week in Review* could include articles from any of the other categories. The main idea of omitting categories was to ensure that the remaining categories were as disjoint as possible. This would create a clearer signal when classifying articles. The following categories were omitted: *Books, Corrections, Front Page, Magazine and Week in Review*.

We also wanted to ensure that the number of articles in each of the selected categories were about the same. To achieve that, we eliminated categories with much fewer articles by selecting the seven categories with the highest number of articles. The seven selected categories are: *Arts, Business, Opinion, Paid Death Notices, Sports, U.S. and World*.

After omitting unwanted categories and selecting the top seven categories with the highest number of articles, we were left with 1.2 million articles. As we did not have enough addressable memory on the machines used for our experiments, the data was then further filtered to include only articles from the last five years. The total number of

articles used for the experiments was 310,000, which is about 18% of the total number of articles. These articles were then formatted into a single file to be consumed by the LDA program.

2. Whitepaper posted on Facebook

The population whitepaper posted on Facebook talks about plans to achieve a sustainable population for the future. The idea of the whitepaper was first mooted due to the declining birth rate in Singapore. It talks about the importance of marriage and parenthood and the measures taken by the government to encourage parenthood. It addresses unpopular immigration policies and speaks about welcoming immigrants while maintaining a strong Singaporean identity. The whitepaper also talks about building a strong workforce and maintaining a strong economy, providing equal opportunities for everyone. The whitepaper elicited a great deal of feedback, often negative. For the present research, we were interested in categorizing the issues or topics of concerns that each piece of feedback received. We extracted comments from the population whitepaper Facebook page and hand annotated the comments into the categories as shown in Table 2.

Marriage and Parenthood
Integration and Identity
Immigrants
Cost of Living
Economy and Workforce
Livability
Others

Table 2. Seven categories of comments on Facebook page.

The categories were chosen as a result of the response the Government of Singapore received from the public with regards to the whitepaper. The feedback from the public could be divided into these seven main categories as noted in the whitepaper feedback website. An example of a comment from each category is shown in Table 3.

Category	Example of comment
Cost of Living	*Hopefully, this means that I can finally buy a flat. Since I'm sandwiched and can't rent due to the income ceiling for HDB rentals.*
Integration and Identity	*That is my point Charlotte. Singaporeans have evolved over time to be distinguished people. Simply giving Singapore passports to "foreign talents" do make them instant Singaporeans in status only. In that context Ive brought up the example of our new citizen/Singaporean Mr Li Yenming who holds a Singapore passport but is a far cry from the distinguished, baked over time Singaporean.*
Economy and Workforce	*This package does not benefit those who earn more than 4000. Super unfair. Why can't the pay increment go by category? This is very unfair. If we earn 4050 and those earn 3500 got increment by 40% their pay will be higher than me as a manager.*
Marriage and Parenthood	*More married women should be encouraged to be pro-family or have a good balance of family and career. This also applies to the married men.*
Immigrants	*KBW, you are nuts! if you have to offer to new citizens the new flats. The criteria should be both foreigners couples have to become new citizen, NOT only one of them! By granting the offer to the couple when only one of them take up the citizenship is a loophole for them to take advantage of Singapore.*
Livability	*now we are packed into cans of sardines in bus and mrt, isn't that enough?? and now u want more sardines into the can again!!!!*

Table 3. Examples of comments in each Facebook category.

While going through the data, we eliminated pure polarity comments that expressed no clear topic. We also eliminated comments that were just hyperlinks and comments that consist of merely neutral quotations. We wanted to discover topics of

concern, so we eliminated comments that did not display any the topic of concern to the Facebook user.

Hand annotating Facebook comments was not a straightforward task. While we understand the context in which a comment was written, it may be a difficult task for a machine to understand these contexts. For example, a user may be posting comments in response to an earlier comment and although the user was concerned with a specific topic as reflected by the earlier comment, it did not surface clearly in his comment itself. Such comments that did not indicate a topic of concern were omitted from the data set.

We also had a lot of comments that were voicing displeasure with the government and were not referring to a specific area of concern with regard to the whitepaper. Some comments were concerned with a range of topics, and some comments were purely insults. As in our experiments, we are only interested in classifying documents into one category each, we picked the most predominant topic of concern of each particular comment.

After removing the unneeded comments, we were left with about 1500 comments. These comments were then formatted into one document to be consumed by the LDA program.

C. REDUCING DIMENSIONS AND MEASURING DISTANCE

There are around 65,000 distinct words found in the NYT corpus and around 9,951 distinct words found in the Facebook corpus. Clustering the data points with such a high dimensionality would be intractable as there is an exponential growth of possible values with every dimension. As dimensionality increases, the volume of the space increases so rapidly that the data becomes very sparse. As a result, the notion of distance between data points becomes less and less significant. Furthermore, at high dimensions, chances are there are many dimensions that are totally unrelated to each other, and hence would appear as noise when clustering.

1. Reducing Dimensions with LDA

We reduced the dimensionality of the data to 50 using the GibbsLDA++ tool [24]. LDA changes the documents from word space into topic space. We then use the normalized topic probability over 50 topics of each document as the new dimensions of the data set. Each document will now have a probability distribution over 50 topics. These 50 topics, however, do not necessarily mean human level topics such as those listed in Table 1 and Table 2.

In LDA, topics are simply described as a probability distribution over a group of words. The topic probability distribution for each document now becomes the feature of the documents on which we will base our clustering algorithm. In our experiments, we use values of 1 and 0.1 for α and β respectively.

2. Cluster Distance Measure

The features used for clustering the documents are the probabilities of the 50 topics per document. Hence instead of using the conventional methods of Euclidean distance or cosine similarity to measure the distance between two data points, we have decided to use the Kullback-Leibler (KL) divergence instead, which measures difference between distributions.

D. CLUSTERING ALGORITHMS

We will be using three different clustering algorithms to conduct our experiments. The first clustering algorithm is the popular K-Means clustering algorithm. We introduce two new methods of clustering using a variable number of clusters, K. It is in our interest to find out if these new methods perform better than standard K-Means clustering.

1. Step K-Means Clustering Algorithm

This algorithm starts K-Means clustering with k equal to one. The algorithm then iteratively increases *k* and clusters until a desired *k* is reached. *k* is increased by the number of outliers present after each round of clustering is completed. The outlying points will be specified as new centroids in addition to the original centroids, and

clustering is done again. The process stops when *k* reaches the desired number, and a final clustering is done.

For example, a clustering starts with only one big cluster whose centroid is randomly chosen. All data points would be clustered to that one centroid, but each with a different KL divergence distance away from the centroid. We then chose the outlying data point and assign it as a new cluster centroid. Now, with two different specified centroids, the clustering is done again. After the clustering is complete, we will have two clusters. The outlying points of both clusters, if any, are then added as new centroids to the original centroid, and clustering is done again. This whole process repeats itself until the number of clusters reaches a desired, pre-specified number. An example of this clustering method is illustrated in Figure 2.

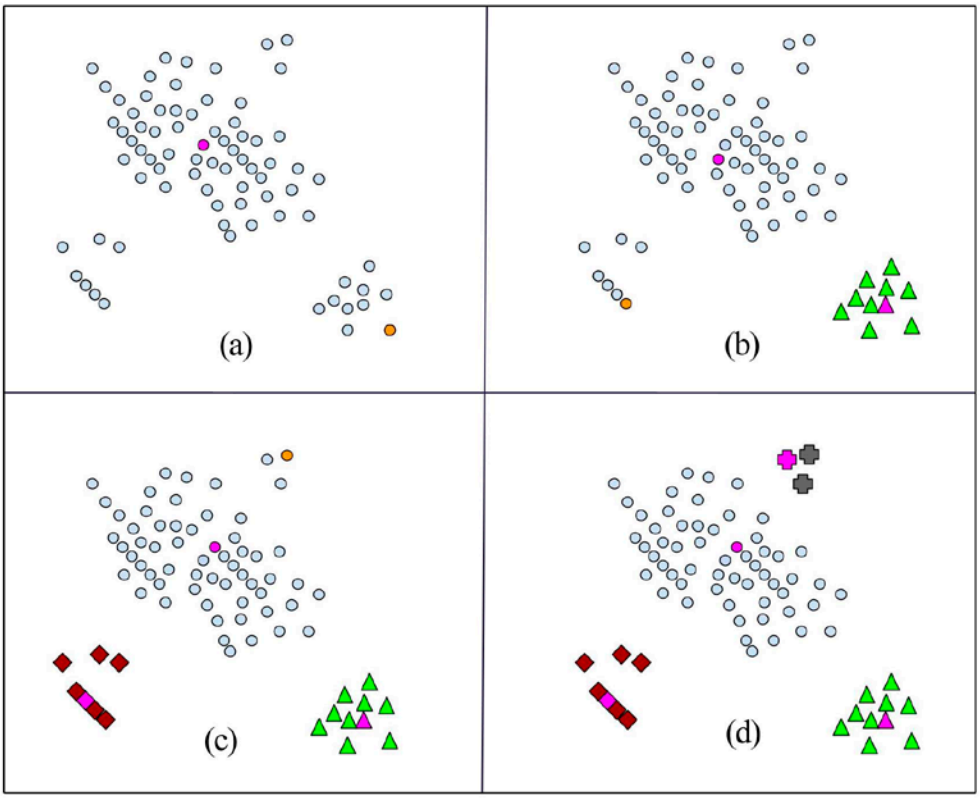

Figure 2. Step K-means clustering with k=4.

In Figure 2(a), an initial centroid is chosen (purple), and an outlier is found (orange). Figure 2(b) shows a second cluster formed with no outliers. The first cluster now has a new outlier. Figure 2(c) shows a third cluster formed with no outliers. The first cluster has now another new outlier. Figure 2(d) shows a fourth cluster formed.

In the algorithm used, we will only choose one outlier per cluster. In the event where there is more than one data point that is specified as an outlier, the algorithm will choose the data point that is furthest away. If the penultimate clustering process produces more outliers than required to reach the desired number of clusters, the algorithm will randomly decide to drop the outliers such that there are only the desired number of clusters in the final clustering process. This algorithm is seen in Figure 3.

```
Algorithm Step-K Means Clustering (Dataset, D)
Begin
1.  k = 1
2.  numClusters =  N //pre-determined cluster limit
3.  centroids = getInitialCentroid(D)
4.  while (k <= N) {
        cluster_result = cluster(D,centroids)
        outliers = findOutliers(cluster_result)
        centroids = centroids + outliers
        k = k + numberOf(outliers)
5.  }
6.  return cluster_result
end
```

Figure 3. Step K-means algorithm.

a. Choosing an Outlier

An outlier is defined as a data point that is considered too far out from the centroid of the cluster to which it is associated. We use standard deviation to determine if a data point can be considered as an outlier. We introduce another parameter, α, to adjust the number of times of the standard deviation a data point can be located from the centroid before it is considered an outlier. Thus, a data point is considered as an outlier if the following condition is met:

$$\text{Distance from centroid } \geq (\alpha \times \sigma) + \mu \qquad (3.1)$$

Where σ is the standard deviation of all the data points in the cluster, and μ is the mean of the distances of all points from the centroid.

b. Choosing the Initial Centroid

Assuming that we know the different categories and the proportion of documents in these categories, we will also attempt to investigate the effect of choosing an initial centroid from a large category as compared to a centroid from a small category.

2. K+J Means Clustering Algorithm

The K+J Means clustering algorithm starts with a K-Means run over a value of *k*. Similar to the Step K-Means clustering algorithm, it then increases the number of clusters by the number of outliers found after the last round of clustering. Clustering only stops when there are no more outliers found. Outliers are found by the same method as described in the Step K-Means clustering algorithm. Similarly, only one outlier per cluster is used in our experiments. This algorithm is depicted in Figure 4.

```
Algorithm K+J Means Clustering (Dataset, D)
Begin
1.  k =  N //pre-determined cluster limit
2.  centroids = getInitialCentroid(D, k)
3.  cluster_result = cluster(D, centroids)
4.  outliers = findOutliers(cluster_result)
5.  while (numberOf(outliers) > 0) {
6.          k = k + numberOf(outliers)
7.          centroids = centroids + outliers
8.          cluster_result = cluster(D,centroids)
9.          outliers = findOutliers(cluster_result)
10. }
11. return cluster_result
end
```

Figure 4. K+J Means algorithm.

The intuition behind this algorithm is that even if we know that we are looking for exactly K topics, we may need J additional clusters because the documents of a given topic may be clustered in different locations. Two important questions with this approach are how we determine what J is and how we determine where the new cluster should be. For K+J means, we answer both of these questions in terms of outliers, because the existence of an outlier is a good indication that that cluster needs to be broken up.

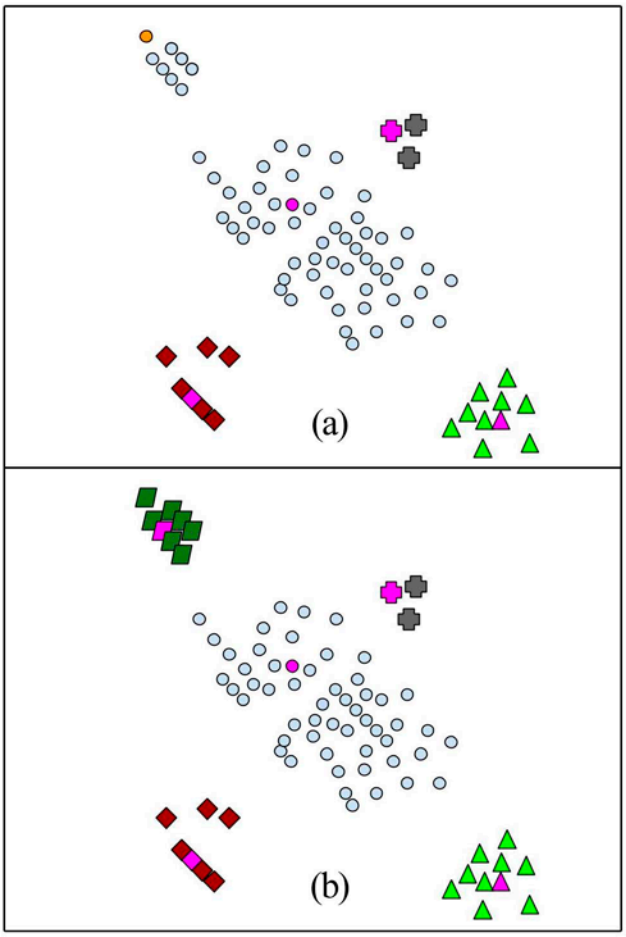

Figure 5. Illustration of K+J Means clustering.

The clustered data when k=4 can be seen in (a). There is an outlier (orange) in the blue circles cluster. As can be seen in Figure 5(b) the algorithm sets the outlier as a new

centroid. A new cluster is formed, and k=5 at this point. No new outliers are detected and the algorithm stops.

3. Classification

For evaluation, we will label a cluster according to 1-Nearest Neighbor: the cluster will inherit the true label of the nearest data point to the cluster centroid. We expect the centroid, and hence the data point nearest to it, to be a good indication of the classification of the cluster. In this procedure, we do not assume clusters to be uniquely labeled, and so it is possible that several topics have no documents labeled under them. Note that the labeling is not actually done by the machine as this is supposed to be an unsupervised method. The labeling and classification are added only in the algorithm to get the classification scores.

E. THE EXPERIMENTS AND EVALUATION CRITERIA

We will be running three experiments with both sets of data. After performing LDA on the data set, we will cluster the data points with the three mentioned clustering algorithms. We will then classify the data points according to the labels of the nearest centroid and compare their results.

True positive, *tp* refers to the number of documents that are correctly identified as a particular category. False positive, *fp* refers to the number of documents that are incorrectly identified as the particular category. True negative, *tn* refers to the number of documents that are correctly identified as not belonging to the particular category. False negative, *fn* refers to the number of documents that are incorrectly identified as not belonging to the particular category. For example, if you are look at category one, *tp* refers to the number of documents that are correctly classified as category one, *fp* would refer to the number of documents that are incorrectly classified as category one, *tn* would refer to the number of documents that are correctly classified as not belonging to category one, and *fn* would refer to the number of documents that should belong to category one, but are incorrectly classified as belonging to another category.

To evaluate the results from the experiments, we will be using accuracy, recall, precision and the F-Score measurement.

1. Accuracy

Accuracy is a common metric to use when performing a multiclass classification. Accuracy is measured as the number of correct classifications to the total data size. We use the following equation to calculate accuracy

$$accuracy = \frac{tp + tn}{tp + fp + tn + fn} \tag{3.2}$$

2. Precision

Precision measures how much of the returned positive is in fact a true positive result. Is it measured with the following equation

$$precision = \frac{tp}{tp + fp} \tag{3.3}$$

3. Recall

Recall is a measure of how much of the true positives are returned. It is measured with the following equation

$$recall = \frac{tp}{tp + fn} \tag{3.4}$$

4. F-Score

F-Score takes into account both precision and recall by finding the harmonic mean of both. It is measured with the following equation

$$fscore = \frac{2 \times precision \times recall}{precision + recall} \tag{3.5}$$

Precision and recall are used more for binary classifiers. In our experiments, we will be using averaged precision and recall values across the different categories.

IV. EXPERIMENT RESULTS AND ANALYSIS

A. OVERVIEW

LDA was used to convert our documents from word space into topic space. We specified a topic space of 50 and used 1 and 0.5 for our LDA parameters α and β, respectively.

We conducted six different experiments based on the three clustering algorithms for both sets of data. As the choice of centroids for the algorithms is random, we ran the clustering programs 10 times to get an average of the accuracy result. We also varied the value of the α parameter in the Step K-Means and K+J Means clustering algorithms to obtain the optimal results.

As we are performing a multiclass classification, the recall and precision values were averaged across the categories for each experiment. The baseline results were also computed to evaluate the performance of the clustering algorithms. The baseline value (MLE) of accuracy is calculated by finding the fraction of the largest category over the total number of documents. The baseline results are tabulated in Tables 4 and 5 for the Facebook and NYT data, respectively.

	Accuracy (MLE)	**Precision**	**Recall**	**F-Score**
Clustering baseline for Facebook data	0.26	0.14	1	0.24

Table 4. Baseline values for the Facebook data.

	Accuracy (MLE)	**Precision**	**Recall**	**F-Score**
Clustering baseline for NYT data	0.23	0.14	1	0.24

Table 5. Baseline values for the NYT data.

B. CLUSTERING ON FACEBOOK DATA

1. K-Means Clustering

Table 6 shows the accuracy results of running K-Means clustering 10 times. We did not specify the initial centroids and allowed the algorithm to select the centroids randomly. The average accuracy of 0.15 falls some way below the MLE value of 0.26. This result reveals the poor performance of K-Means clustering and classification of documents using topics as features.

Iteration	Accuracy
1	0.118609
2	0.099523
3	0.122018
4	0.128153
5	0.133606
6	0.141104
7	0.152011
8	0.164963
9	0.173142
10	0.234492
Average	0.146762

Table 6. Table of accuracy scores for K-Means clustering on Facebook data.

2. Step K-Means Clustering

Table 7 shows the results of Step K-Means clustering with the value α varying from 1.1 to 4. The clustering is done 10 times with each value of α, and the average is

shown here. The accuracy peaked at 0.176551 when the value of α was 4. We can consistently obtain this accuracy by keeping the value of α at 4.

α	Accuracy
1.1	0.174506
1.2	0.175869
1.3	0.174506
1.4	0.174506
1.5	0.175187
2	0.175187
2.1	0.175869
2.2	0.175869
2.3 to 3.9	0.176551
4	0.176551
Average	0.176005

Table 7. Step K-Means clustering accuracy with varying α.

The average accuracy of 0.18 also falls below the MLE value. This demonstrates that Step K-Means clustering also does not do a good job of classifying the documents. However, it is worthy to note that despite starting with random centroids and varying α, the accuracy remains largely the same. The standard deviation of the values in Step K-Means is 0.0008 compared to 0.0379 in K-Means.

3. K+J Means clustering

Table 8 shows a subset of the results of K+J Means clustering with α varying between 1.1 to 1.8 and the proportion of outliers added as new clusters from 20% to 100%. The proportion parameter gives us another parameter to adjust as we are trying to

find cases where not all outliers need to be added to achieve the same level of accuracy. We have also limited the number of clusters to not exceed 30% of the total number of documents.

α	Proportion	Resulting number of clusters	Accuracy
1.8	50%	303	0.364938608
1.1	50%	305	0.327421555
1.5	25%	310	0.344474761
1.4	20%	315	0.342428377
1.7	100%	322	0.369713506
1.7	50%	325	0.371077763
1.5	25%	335	0.360845839
1.3	20%	337	0.36425648
1.3	20%	337	0.36425648
1.4	25%	387	0.392905866
Average			0.36

Table 8. K+J Means clustering accuracy.

The average accuracy of 0.36 is a result of increasing the number of clusters within the permitted range. A quick run of K-Means clustering with 387 clusters gives an accuracy of 0.36 on the average, whereas K+J Means clustering gives an accuracy of 0.39 consistently.

4. Precision, recall and F-score

Table 9 shows the precision, recall and f-score of all three clustering algorithms.

	Accuracy	Precision	Recall	F-Score
K-Means	0.15	0.13	0.18	0.15
Step K-Means	0.18	0.192	0.15	0.185
K+J Means (with an average of 344 clusters)	0.36	0.543	0.359	0.43

Table 9. Summary of results of all three clustering algorithms on Facebook data.

The results show that K-Means and Step K-Means fall below the MLE for accuracy and are therefore not good at classifying Facebook documents in topic space. Step K-Means does slightly better than baseline with precision, but the improvement is not significant. K+J Means does considerably better than the baseline scores due to the fact that it is clustering with over 300 clusters. However, when compared to K-Means at over 300 clusters, the improvement is consistent, albeit slight.

C. CLUSTERING ON NYT DATA

1. K-Means Clustering

Table 10 shows the accuracy result of K-Means clustering on the NYT data over 10 iterations. As the starting centroids were chosen at random, the accuracy score of each iteration differs. We then calculated the average of the accuracy values. The average accuracy of 0.4955 is considerably better than the MLE value of 0.23. This result shows that there is some signal in topic space of the NYT data.

Iteration	Accuracy
1	0.4491
2	0.4915
3	0.5242
4	0.4925
5	0.5034
6	0.4427
7	0.5264
8	0.5416
9	0.5323
10	0.4512
Average	0.4955

Table 10. K-Means clustering result on NYT data.

2. Step K-Means Clustering

Table 11 shows the result of Step K-Means clustering on the NYT data. The α parameter was varied from 1 to 2.3. The peak accuracy was found when the value of α was at 1.3 and 1.5. If we keep the value of α at those values, the accuracy results will consistently be around 0.5317. The accuracy score of 0.5317 is considerably better than the MLE value of 0.23 and better than the K-Means clustering value of 0.4955.

α	Accuracy
1	0.5066
1.1	0.5122
1.3	0.5317
1.4	0.5122
1.5	0.5317
1.7	0.5066
2	0.5122
2.1	0.5122
2.2	0.5122
2.3	0.5122
Average	0.515

Table 11. Step K-Means clustering result on NYT data with varying α.

3. K+J Means clustering

Table 12 shows the result of K+J Means clustering on the NYT data with the value of α varied between 1.1 and 2. We capped the maximum number of clusters to 200, which is about six percent of the total number of documents. The accuracy score is highest when α is at a value of 2. This accuracy score of 0.81 can be consistently achieved if the alpha is fixed at 2, and it is significantly higher than the MLE value of 0.23. This is partly due to the significant increase in the number of clusters. A quick run of K-Means clustering using 200 clusters gives an accuracy score of about 0.77.

α	Resulting number of clusters	Accuracy
1.1	200	0.8
1.2	200	0.78
1.3	200	.79
1.4	200	.8
1.5	200	0.79
1.6	200	0.8
1.7	200	0.79
1.8	200	0.78
1.9	200	0.79
2	200	0.81
Average		0.79

Table 12. Accuracy result of K+J Means clustering.

4. Precision, Recall and F-score

The average precision, recall and F-score of all three clustering algorithms on the NYT data are tabulated in Table 13.

	Accuracy	Precision	Recall	F-Score
K-Means	0.49	0.35	0.44	0.39
Step K-Means	0.53	0.43	0.42	0.42
K+J Means (with an average of 200 clusters)	0.81	0.79	0.79	0.79

Table 13. Summary of results of all three clustering algorithms on NYT data.

This result shows that the NYT data contains far more signal than the Facebook data. The accuracy, precision and recall scores of all the clustering algorithms are higher than the baseline values. Step K-Means performed about four percent better than K-Means in terms of accuracy. K+J Means clustering obtained the best results with an accuracy of 0.81 with 200 clusters.

D. ANALYSIS OF RESULTS

1. Facebook Data

The clustering methods produced poor results when attempting to cluster the Facebook data in document space. One hypothesis to explain this poor result could be the presence of inherent noise in the data set due to the fact that social media comments are usually short and may not have much signal to begin with.

The results show that Step K-Means clustering does about 3% better than K-Means consistently, which means it managed to classify about 40 more data points correctly. The reason for this consistency is the way the centroids are chosen. In Step K-Means and K+J Means, cluster outliers are chosen as new centroids as outliers have a better chance of belonging to another category, and hence another cluster. This method of choosing new centroids avoids converging into a local minimum to which K-Means is susceptible.

We ran the same experiment over six categories and compared our results to Phua's result [20]. Phua used Labeled LDA (LLDA) to classify the same set of documents over six categories instead of seven. She eliminated the *others* category. Table 14 highlights the difference in results.

	Accuracy	Precision	Recall	F-Score
K-Means	0.12	0.11	0.15	0.13
Step K-Means	0.11	0.43	0.17	0.24
K+J Means (with an average of 366 clusters)	0.36	0.61	0.4	0.48
LLDA	0.45	0.4	0.41	0.4

Table 14. Comparison of results using clustering methods and LLDA.

LLDA outperformed the clustering methods predictably as it is a supervised method. LLDA builds a model of the truth while learning from a training set of data and then constrains the distribution of topics in the test data to match the learned models. It is interesting to note that the precision score for Step K-Means and K+J Means are higher than that for LLDA. This is because the scores here are averaged across all the categories. There are certain categories that had zero recall and precision, and other categories with very high recall and precision. This phenomenon occurs because in our clustering algorithms, we label data points as the label of the data point that is nearest to its nearest centroid, and there is a chance that not all categories are represented in the centroids of the clusters.

2. *New York Times* Data

We used the NYT corpus containing articles that are generally long and coherent to a specific topic. It is very well annotated, and hence, LDA should perform well with this data. Zhao and Jiang [21] managed to obtain meaningful topics using LDA on the NYT data, although no measure was provided on how well the topics were discovered.

The results of our experiments show the algorithms work much better on the NYT data than on the Facebook data. All three clustering algorithms performed significantly better than the baseline, and Step K-Means performed about 4% better than K-Means. This means that Step K-Means was able to classify correctly about 12,400 articles more than K-Means could. Similarly, K+J Means clustering was able to obtain an accuracy of 0.81 with 200 clusters, about 3% better than K-Means with 200 clusters. As mentioned

previously, the slight increase in performance was due to the way the centroids were chosen.

E. SUMMARY

We found that the longer and more topic coherent a data set is, the better the performance of the clustering algorithms. This is due to the fact that a topic inference algorithm such as LDA is used to reduce the dimensionality of the data before the clustering algorithms are applied.

In our experiments, we hypothesize that the result on the Facebook data was poor because the data, like most social media content, is short and not particularly topic coherent.

The NYT data, on the other hand, is a well curated and annotated corpus. LDA did a good job in changing the features from word space to topic space, and as a result, the clustering algorithms worked better.

Hautamäki *et al*. [22] improved clustering performance by removing the outliers. However, removing outliers also meant losing data, and hence, while removing outliers may work in other fields, such as image processing, it is not suitable in the field of document classification. The Step K-Means and K+J Means clustering algorithms both rely on finding outliers to improve their performance. They make use of the assumption that having centroids from different categories improves the performance of the clustering. By choosing outliers as the centroids, we are assuming that cluster outliers belong to a different category y and hence should belong to another cluster.

By classifying the data points as the category of the nearest centroid may also result in an entire category not being labeled. Perhaps, another way of labeling data points is by looking at the classification of the all the data points in the cluster instead of the centroid only. A way of doing this is perhaps through the use of active learning. By showing a user a sample of data points in a cluster, the user may be able to give a more accurate classification of the cluster as compared to just looking at the cluster centroid. Note that the labeling is not actually done by the machine as this is supposed to be an

unsupervised method. The labeling and classification are added only in the algorithm to get the classification scores.

Although the results were not very convincing, there is some evidence that with some enhancements, the Step K-Means and K+J Means algorithms may have the ability to perform much better than the normal K-Means algorithm.

V. FUTURE WORK AND CONCLUSION

A. FUTURE WORK

1. Active Learning Semi-Supervision Clustering

Currently our algorithm picks outliers based on its distance away from the centroid. It then immediately assumes that the outlier should belong to another cluster. Instead of assuming, we can ask if the outlier belongs to the cluster. This active learning component enables the algorithm to select its outliers more effectively which is important in improving the performance of the classification.

2. Allow More Than One Outlier Per Cluster

With active learning, it would also be easier to implement the ability to have more than one outlier per cluster. It is highly possible that a cluster may have more than one valid outlier as illustrated in Figure 6; our current algorithm will only pick the one furthest away from the centroid as an outlier. With active learning, we can ask questions of both outliers before deciding if they should belong to new clusters.

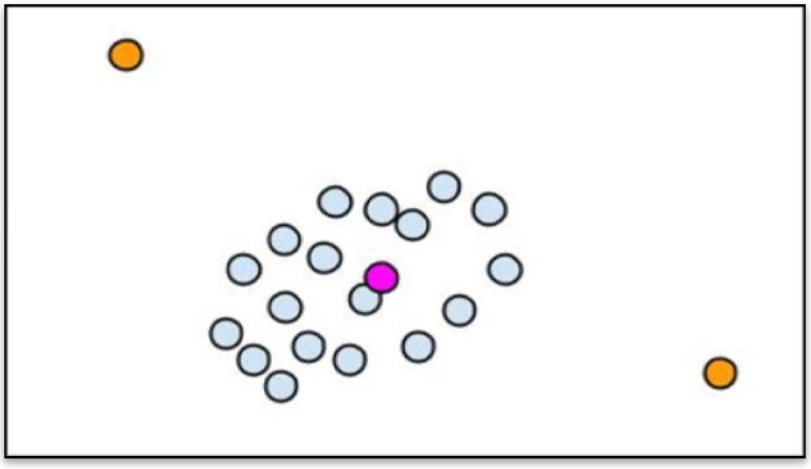

Figure 6. Illustrating two valid outliers in a cluster.

3. Constrained Clustering

As discussed earlier, Basu *et al.* [14] described a way to perform constrained clustering by adding a cost for violating a known pairwise rule to the objective function. Following on from the idea of asking if a data point is indeed an outlier, we can then constrain that data point to a particular category once we know its true label. As more questions are asked, more constraints can be added. Clustering is now done by ensuring that at every clustering iteration, there will not be a cluster with data points belonging to different categories. We could allow for violations if the clustering does not converge by assigning a cost for violating a constraint and having the algorithm minimize that cost. Constrained clustering does not merely look at the distance metric of data points, but also at the cost, if any, of violating a constraint.

4. Selecting Clusters to Split

Savaresi et al. [19] described a method to select a cluster to split based upon the shape of the cluster. If we can use a similar method to identify clusters that need to be split, we would not have to look at all clusters. This would save processing time when processing large data sets. It would also be interesting to investigate the relationship between a pure cluster (a cluster that contains mostly data points belonging to the same category) and the shape of that cluster.

5. Finding Optimal Values of the Clustering Parameter, α

As we have seen, the value of α affects the number of outliers detected. Currently to find the optimal α, we have to run the clustering algorithm repeatedly with different values of α. It is worth the effort to explore finding the optimal α for a cluster by looking at the characteristics of the cluster. Characteristics could include size, shape and density. Knowing the correct value of α to use and the ability to have different α values for different clusters would eliminate the need to run the same experiment repeatedly and would also greatly improve the accuracy of the clustering.

6. Choosing a Larger and Better Data Set

We hypothesize that the reason for the poor classification results was because of the inherent noise in the data set. We should try using a different set of social media data and exclude more stop words that are unique to Singlish. We should also use a much larger data set than what was used in our experiments. As far as possible, we should also select data that is more topic coherent.

7. Running the Experiment on NYT Abstracts

Instead of running the experiments on the NYT articles, we can run the experiments on the article abstracts. This would represent a data set that is more similar to the Facebook data set in terms of document length. This would eradicate the length of document as a reason for better signal in the NYT data.

B. CONCLUSION

Our experiments did poorly with the Facebook data, and quite possibly would with any other data that do not show topic coherence because we used LDA to reduce data dimensionality before the clustering. However, a marked improvement in the clustering results for the NYT data show that the combination of LDA and clustering is able to perform decently as a classifier.

In addition, we confirmed that Step K-Means and K+J Means both performed slightly better than K-Means. The choice of cluster outliers as new cluster centroids is key to the algorithm. However, there is much room for improvement in selecting outliers, and perhaps with enhancements as mentioned as future work, it would perform much better as a classifier.

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX. CONFUSION MATRICES

		Truth							
		cat1	cat2	cat3	cat4	cat5	cat6	cat7	
Prediction	cat1	4	6	7	7	11	8	0	**43**
	cat2	0	0	0	0	0	0	0	**0**
	cat3	5	3	24	32	24	16	2	**106**
	cat4	6	12	15	21	10	14	0	**78**
	cat5	12	27	35	47	48	32	17	**218**
	cat6	0	0	0	0	0	0	0	**0**
	cat7	34	88	152	271	174	185	118	**1022**
		61	**136**	**233**	**378**	**267**	**255**	**137**	

Table 15. Confusion Matrix of K-Means on Facebook data.

		Truth							
		cat1	cat2	cat3	cat4	cat5	cat6	cat7	
Prediction	cat1	4	8	7	7	8	6	0	**40**
	cat2	0	0	0	0	0	0	0	**0**
	cat3	0	0	5	1	0	1	0	**7**
	cat4	0	0	0	0	0	0	0	**0**
	cat5	1	0	3	1	5	4	0	**14**
	cat6	56	128	218	369	254	244	137	**1406**
	cat7	0	0	0	0	0	0	0	**0**
		61	**136**	**233**	**378**	**267**	**255**	**137**	

Table 16. Confusion Matrix of Step K-Means on Facebook data.

		Truth							
		cat1	cat2	cat3	cat4	cat5	cat6	cat7	
Prediction	cat1	28	4	6	8	5	4	1	**56**
	cat2	1	35	4	5	2	2	0	**49**
	cat3	22	75	195	228	134	157	111	**922**
	cat4	3	9	16	114	18	21	7	**188**
	cat5	3	4	3	15	94	10	3	**132**
	cat6	3	6	7	8	13	55	3	**95**
	cat7	1	3	2	0	1	6	12	**25**
		61	**136**	**233**	**378**	**267**	**255**	**137**	

Table 17. Confusion Matrix of K+J Means on Facebook data.

		Truth							
		cat1	cat2	cat3	cat4	cat5	cat6	cat7	
Prediction	cat1	67402	143	112	13	267	210	39	**68186**
	cat2	18	34624	6576	14674	5301	11525	22842	**95560**
	cat3	3401	21636	27539	15239	9216	22898	1317	**101246**
	cat4	0	0	0	0	0	0	0	**0**
	cat5	34	6885	4328	8557	22500	1794	250	**44348**
	cat6	0	0	0	0	0	0	0	**0**
	cat7	0	0	0	0	0	0	0	**0**
		70855	**63288**	**38555**	**38483**	**37284**	**36427**	**24448**	

Table 18. Confusion Matrix of K-Means on NYT data.

		Truth							
		cat1	cat2	cat3	cat4	cat5	cat6	cat7	
Prediction	cat1	60929	0	153	183	53	74	27	**61419**
	cat2	0	0	0	0	0	0	0	**0**
	cat3	39	88	13756	5	0	3542	20	**17450**
	cat4	0	0	0	0	0	0	0	**0**
	cat5	0	227	404	63	31145	1027	72	**32938**
	cat6	9887	38168	22971	36176	7357	58645	24329	**197533**
	cat7	0	0	0	0	0	0	0	**0**
		70855	**38483**	**37284**	**36427**	**38555**	**63288**	**24448**	

Table 19. Confusion Matrix of Step K-Means on NYT data.

		Truth							
		cat1	cat2	cat3	cat4	cat5	cat6	cat7	
Prediction	cat1	68598	172	349	279	36	559	105	**70098**
	cat2	1831	20827	1788	1424	524	6947	571	**33912**
	cat3	191	1990	31256	1621	479	4867	853	**41257**
	cat4	18	2256	229	29683	1017	2414	986	**36603**
	cat5	4	394	100	159	34904	1168	59	**36788**
	cat6	200	7218	3250	2117	1374	45914	3759	**63832**
	cat7	13	5626	312	1144	221	1419	18115	**26850**
		70855	**38483**	**37284**	**36427**	**38555**	**63288**	**24448**	

Table 20. Confusion Matrix of K+J Means on NYT data.

		Truth							
		cat1	cat2	cat3	cat4	cat5	cat6	cat7	
Prediction	cat1	70727	2646	1070	1163	968	759	166	
	cat2	37	45357	1063	4955	5697	2330	4693	
	cat3	0	852	35161	483	113	1066	54	
	cat4	2	3620	270	21243	1376	2652	1820	
	cat5	67	3794	203	2465	27951	960	403	
	cat6	20	4342	559	1872	782	27928	866	
	cat7	2	2677	229	6302	397	732	16446	
		70855	**63288**	**38555**	**38483**	**37284**	**36427**	**24448**	

Table 21. Confusion Matrix of K Means on NYT data with k=200.

		Truth						
		cat1	cat2	cat3	cat4	cat5	cat6	
Prediction	cat1	14	34	36	64	48	54	**250**
	cat2	33	52	111	182	131	118	**627**
	cat3	0	0	4	1	0	0	**5**
	cat4	0	0	0	0	0	0	**0**
	cat5	9	45	68	123	75	72	**392**
	cat6	5	5	14	8	13	11	**56**
		61	**136**	**233**	**378**	**267**	**255**	

Table 22. Confusion Matrix of K-Means on Facebook data with 6 categories.

		Truth						
		cat1	cat2	cat3	cat4	cat5	cat6	
Prediction	cat1	0	0	0	0	0	0	**0**
	cat2	56	124	215	363	247	241	**1246**
	cat3	3	12	16	15	13	11	**70**
	cat4	0	0	0	0	0	0	**0**
	cat5	0	0	0	0	2	0	**2**
	cat6	2	0	2	0	5	3	**12**
		61	**136**	**233**	**378**	**267**	**255**	

Table 23. Confusion Matrix of Step K-Means on Facebook data with 6 categories.

		Truth						
		cat1	cat2	cat3	cat4	cat5	cat6	
Prediction	cat1	24	1	2	7	1	2	**37**
	cat2	25	114	149	236	144	168	**836**
	cat3	3	7	63	10	11	3	**97**
	cat4	3	7	5	108	9	5	**137**
	cat5	4	3	3	7	96	9	**122**
	cat6	2	4	11	10	6	68	**101**
		61	**136**	**233**	**378**	**267**	**255**	

Table 24. Confusion Matrix of K+J Means on Facebook data with 6 categories.

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF REFERENCES

[1] J. Allen, J. Carbonell, G. Doddington, J. Yamron, and Y. Yang, "Topic detection and tracking pilot study: Final report," *DARPA Broadcast News Transcription and Understanding Workshop*, Lansdowne, Virginia, February 8–11, 1998.

[2] Singapore National Population and Talent Division. "A Sustainable Population for a Dynamic Singapore: Population White Paper," 2013.

[2] D.M. Blei, A. Ng, M. Jordan, "Latent Dirichlet Allocation," *Journal of Machine Learning Research*, vol. 3, pp. 993–1022, 2003.

[3] C. Wartena, and R. Brussee, "Topic detection by clustering keywords," *DEXA '08: Proceedings of the 2008 19th International Conference on Database and Expert Systems Application*, Turin, Italy, 2008.

[4] T. Hofmann, "Probabilistic latent semantic indexing," in *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 50-57, Berkeley, California, August 15–19, 1999.

[5] D. Baker, T. Hoffman, A. McCallum, and Y. Yang, "A hierarchical probabilistic model for novelty detection in text," 1999.

[6] E. Eskin, W. Lee, and S. Stolfo, "Modeling system call for intrusion detection using dynamic window sizes," in *Proceedings of DISCEX*, Anaheim, California, June 12–14, 2001.

[7] D. E. Denning, "An intrusion detection model," *IEEE Trans. Softw. Eng.*, vol. 13, pp. 222–232, 1987.

[8] T. Fawcett, F. Provost, "Activity monitoring: Noticing interesting changes in behavior," in *Proceedings of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 53-62, San Diego, California, USA, August 15–18, 1999.

[9] C. Cortes, and V. Vapnik, "Support-vector network," *Machine Learning*, vol. 20, pp. 1–25, 1995.

[10] S. W. Hilt, and S. Merat, "SVM clustering," *BMC Bioinformatics*, vol. 8, Suppl. 7, S18, 2007.

[11] A. K. Jain, and R. C. Dubes, *Algorithms for Clustering Data*, Upper Saddle River, NJ: Prentice-Hall, 1988.

[12] J. Hartigan, and M. A. Wong, "A K-means clustering algorithm," *Applied Statistics* 28, pp. 100–108, 1979.

[13] S. Kullback, and R. A. Leibler, "On information and sufficiency," *Annals of Mathematical Statistics*, vol. 22, no. 1, pp. 79–86, 1951.

[14] S. Basu, A. Banerjee, and R. J. Mooney, "Active semi-supervision for pairwise constrained clustering," in *Proceedings of the SIAM International Conference on Data Mining*, Nashville, Tennessee, June 13–16, 2004.

[15] H. P. Kriegel, P. Kroger, and A. Zimek, "Outlier detection techniques (tutorial)," *13th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD),*Bangkok, Thailand, April 27–30, 2009.

[16] H. P. Kriegel, P. Kroger, and A. Zimek, "Clustering high-dimensional data: A survey on subspace clustering, pattern based clustering, and correlation clustering," *ACM Transactions on Knowledge Discovery from Data (TKDD*), vol. 3, p. 1–58, 2009.

[17] H. Hotelling, "The most predictable criterion," *J. Ed. Psych.,* vol. 26, pp. 139–142, 1935.

[18] C. Ding, X. He, "K-means clustering via principal component analysis," in *Proceedings of the Twenty-first International Conference on Machine Learning*, Banff, Canada, 4–8 July, 2004.

[19] S. M. Savaresi, D. L. Boley, S. Bittanti, and G. Gazzaniga, "Choosing the cluster to split in bisecting divisive clustering algorithms," *Second SIAM International Conference on Data Mining,* Arlington, Virginia, April 11–13, 2002.

[20] Y. L. Phua, "Social media sentiment analysis and topic detection for Singapore English," Master's Thesis, Department of Computer Science, Naval Postgraduate School, Monterey, CA, 2013.

[21] X. Zhao, and J. Jiang, "An empirical comparison of topics in twitter and traditional media," Singapore Management University School of Information Systems, Technical Paper Series, Singapore Management University, Singapore, January 20, 2011.

[22] V. Hautamäki, S. Cherednichenko, I.Kärkkäinen, T. Kinnunen, and P. Fränti, "Improving K-means by outlier removal," in *SCIA'05 Proceedings of the 14th Scandinavian Conference,* Joensuu, Finland, June 19–22, 2005.

[23] E. Sandhaus, "The New York Times Annotated Corpus" [Online] 2008, Available: http://www.ldc.upenn.edu/Catalog /catalogEntry.jsp?catalogId=LDC2008T19. [Accessed: August 1, 2013].

[24] X. H. Phan and C. T. Nguyen, "A C/C++ implementation of latent Dirichlet allocation (LDA)" [Online] 2007, Available: http://gibbslda.sourceforge.net. [Accessed: August 1, 2013].

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California